CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A security module for use with a terminal, comprising

a data interface adapted to be coupled to a terminal, for receiving one of part of an algorithm code and the complete algorithm code from the terminal, with the algorithm code concerning a processing of secrets;

a power interface for receiving ~~supply~~ power from the terminal;

a volatile memory for storing the one of the part of the algorithm code and the complete algorithm code received via the data interface, said volatile memory being coupled to the power interface in order to have power supplied thereto such that said volatile memory will be cleared upon an interruption of the receipt of the ~~supply~~ power from the terminal; and

a processor for performing the algorithm code in order to obtain an algorithm code result that can be delivered to the terminal.

Claim 2 (currently amended). A security module according to claim 1, wherein the data interface is adapted to receive only the part of the algorithm code, the security module further comprising:

a non-volatile memory in which a remainder of the algorithm code ~~is stored along with the received part of the algorithm code for forming the complete algorithm code~~ which, along with the received part of the algorithm code, forms the complete algorithm code, is stored.

Claim 3 (original): A security module according to claim 1, further comprising:

a means for performing an authentication between the terminal and the security module.

Claim 4 (previously presented): A security module according to claim 1, wherein the data interface is arranged to receive from the terminal the one of the part of the algorithm code and the complete algorithm code in encrypted form and with a certificate, with the security module further comprising:

a means for decrypting the one of the part of the encrypted algorithm code and the encrypted complete algorithm code; and

a means for examining the certificate and for preventing performing of the algorithm code depending on examination of said certificate.

Claim 5 (previously presented): A security module according to claim 1, wherein said data interface is adapted to receive only the part of the algorithm code, the security module further comprising:

a memory managing unit for controlling memory accesses of the processor, with the part of the algorithm code containing addresses of the algorithm code.

Claim 6 (currently amended): A security module according to claim 1, further comprising:

a means for monitoring a predetermined security condition and for clearing the volatile memory if said predetermined security condition is fulfilled, with said security condition being selected from ~~a plurality of conditions comprising~~ an interruption of a supply voltage, ~~an irregularity and~~ a fluctuation of the supply voltage and an interruption of a system clock ~~as well as of additional operating parameters~~.

Claim 7 (currently amended): A security module according to claim 1, wherein the algorithm code comprises a program code selected for carrying out a task selected from [a] the group consisting of a symmetric cryptographic algorithm, an asymmetric cryptographic algorithm, an RSA algorithm, a cryptographic

process according to the DES standard, an elliptic curve process, ~~and~~ an access function for accessing a digital value stored on the security module ~~as well as~~ and an access function for changing the digital value stored on the security module.

Claim 8 (currently amended): A security module according to claim 1, wherein said data interface is adapted to receive only the part of the algorithm code, the part of the algorithm code ~~comprising~~ consisting of one or more of a start address of the algorithm code, memory addresses of computing components necessary for performing the algorithm code, ~~or~~ and jump addresses of the algorithm code.

Claim 9 (currently amended): A security module according to claim 1, wherein the data interface is adapted to receive the one of the part of the algorithm code and the complete algorithm code several times in different versions, with the volatile memory being arranged ~~for storing a newly received version~~ for being overwritten by different versions of one of the part of the of the algorithm code and the complete algorithm code ~~such that the previously received version of one of the part of the algorithm and the complete algorithm code is overwritten~~ at the several times.

Claim 10 (previously presented): A security module according to claim 1, wherein said security module is designed as a chip card.

Claim 11 (currently amended): A process for computing an algorithm code result using a security module, comprising the steps of:

receiving one of part of an algorithm code and the complete algorithm code by means of an interface to a terminal, with the algorithm code concerning a processing of secrets;

storing the one of the part of the algorithm code and the complete algorithm code in a volatile memory of the security module, with the volatile memory being coupled to the interface, to be supplied with power, such that the volatile memory will be cleared upon an interruption of the receipt of the ~~supply~~ power from the terminal;

performing said algorithm code on the security module in order to obtain an algorithm code result; and

delivering said algorithm code result to the terminal.

Claim 12 (currently amended): A process according to claim 11, further comprising removing the security module from the terminal thereby causing an interruption of the receipt of the ~~supply~~ power to the volatile memory from the terminal and clearing said volatile memory upon interruption of the receipt of ~~supply~~ power from the terminal.

Claim 13 (currently amended): A terminal for use with a security module, comprising:

a data interface adapted to be coupled to the security module, for transmitting at least part of an algorithm code or the complete algorithm code from the terminal to a volatile memory of the security module and for receiving an algorithm code result from the security module, with the algorithm code concerning a processing of secrets; and

a power interface for delivering ~~supply~~ power to the security module, with the volatile memory being supplied by the ~~supply~~ power, such that the volatile memory will be cleared upon an interruption of the receipt of the ~~supply~~ power from the terminal,

~~with the terminal, for each communication operation between terminal and security module, being adapted to, during a single one of the communication operations with the security module,~~
for each communication operation between the terminal and the security module, the data interface controlled to: send at least the part of the algorithm code or the complete algorithm code to the volatile memory of the security module~~; and, after sending,~~ and then to receive the algorithm code result from the security module.

Claim 14 (currently amended): A process for controlling ~~within a plurality of communication operations,~~ a security module using a terminal in order to obtain an algorithm code result from the security module, ~~with~~ the process comprising ~~for each communication operation,~~ performing the following steps during ~~a single one of the~~ each one of a plurality of communication operations ~~with~~ between the terminal and the security module:

delivering ~~supply~~ power from the terminal to the security module;

transmitting at least part of an algorithm code or the complete algorithm code from the terminal to a volatile memory of the security module, with the algorithm code concerning a processing of secrets, with the volatile memory being supplied by the ~~supply~~ power, such that the volatile memory will be cleared upon an interruption of the receipt of the ~~supply~~ power from the terminal; and

receiving an algorithm code result from the security module.

Claim 15 (currently amended): A process for communication between a security module and a terminal, comprising the steps of:

transferring one of part of an algorithm code and the complete algorithm code from the terminal to the security module, with the algorithm code concerning a processing of secrets;

storing the one of the part of the algorithm code and said complete algorithm code in a volatile memory of the security module, with the volatile memory being supplied by ~~supply~~ power from the terminal, such that the volatile memory will be cleared upon interruption of the receipt of the ~~supply~~ power from the terminal;

performing said algorithm code on the security module in order to obtain an algorithm code result;

delivering said algorithm code result to the terminal; and

clearing said volatile memory upon an interruption of the receipt of the ~~supply~~ power from the terminal.

Claim 16 (currently amended): A process according to claim 15, further comprising:

sequentially transferring a plurality of different versions of the one of the part of the algorithm code and said complete algorithm code; and

sequentially storing ~~the~~ the different versions of the one of the part of the algorithm code and the complete algorithm code such that a respective previous version of the plurality of different versions of the one of the part of the algorithm code and the complete algorithm code is overwritten.

Claim 17 (currently amended): A security module for use with a terminal, comprising:

a data interface adapted to be coupled to a terminal, for receiving ~~a first~~ part of an algorithm code from the terminal, with the algorithm code concerning a cryptographic processing of data;

a power interface for receiving ~~supply~~ power from the terminal;

a volatile memory for storing the ~~first~~ part of the algorithm code received via the data interface, said volatile memory being coupled to said power interface in order to have power supplied thereto such that the volatile memory will be cleared upon an interruption of the receipt of the ~~supply~~ power from the terminal;

a non-volatile memory in which a ~~second part~~ remainder of the algorithm code ~~which is a non-received remainder the algorithm code is stored~~ which, along with the received part of the algorithm code, forms a complete algorithm code, is stored; and

a processor for performing the algorithm code in order to obtain cryptographically processed data that can be delivered to the terminal, wherein the ~~first~~ part of the algorithm code includes memory addresses of computing

components necessary for performing the algorithm code, or jump addresses of the algorithm code pointing to partial routines of the algorithm code.